CLAIMS

1. (Currently Amended) A method for tagging an allocable memory block, comprising:

determining the identity of a routine performing one of requesting the allocable memory block, requesting the size of the allocable memory block, and freeing the allocable memory block;

generating an identifier for the routine;

storing the identifier in the allocable memory block; and

storing a timestamp within the allocable memory block, wherein the timestamp is configured to indicate ~~indicates~~ a time when ~~one of the requesting and the freeing of the allocable memory block is performed~~:

the requesting of the allocable memory block is performed unless the timestamp indicates a time when the allocable memory block is freed; and

the freeing of the allocable memory block is performed unless the timestamp indicates a time when the allocable memory block is requested,

such that, upon detection of a memory usage error involving the allocable memory block, the identifier for the routine and the timestamp provide information usable in determining whether the routine is causing memory errors.

2. (Original) The method of Claim 1, further comprising examining the heap to determine the presence of memory errors.

3. (Original) The method of Claim 1, further comprising performing a checksum on the allocable memory block and storing the results of the checksum within the allocable memory block.

4. (Original) The method of Claim 3, further comprising examining the results of the checksum to determine the presence of memory errors.

5. (Original) The method of Claim 1, wherein the identifier is the return address of the identified routine.

6. (Original) The method of Claim 1, further comprising writing a memory overwrite detection pattern within the allocable memory block.

7. (Original) The method of Claim 6, wherein the memory overwrite detection pattern is written within an area of the allocable memory block that is used for alignment purposes.

8. (Original) The method of Claim 1, wherein an identifier is generated and stored for a routine that requests the allocable memory block and an identifier is generated and stored for a routine that frees the memory block.

9. (Original) The method of Claim 1, further comprising storing a heap index for the allocable memory block within the allocable memory block, wherein the heap index points to one of a plurality of heaps.

10. (Canceled).

11. (Currently Amended) A computer-readable medium having computer-executable components for tagging an allocable memory block, comprising:

a first component that is arranged to determine the identity of a routine performing one of requesting the allocable memory block, requesting the size of the allocable memory block, and freeing the allocable memory block;

a second component that is arranged to generate an identifier for the routine;

a third component that is arranged to store the identifier in the allocable memory block; and

a fourth component that is arranged to store a timestamp within the allocable memory block, wherein the timestamp is configured to indicate ~~indicates~~ a time when ~~at which the routine performs one of the requesting and the freeing of the allocable memory block~~:

the requesting of the allocable memory block is performed unless the timestamp indicates a time when the allocable memory block is freed; and

the freeing of the allocable memory block is performed unless the timestamp indicates a time when the allocable memory block is requested,

such that, upon detection of a memory usage error involving the allocable memory block, the identifier for the routine and the timestamp provide information usable in determining whether if the routine is causing memory errors.

12. (Original) The computer-readable medium of Claim 11, further comprising an examination component that is arranged to examine the heap to determine the presence of memory errors.

13. (Original) The computer-readable medium of Claim 12, further comprising a checksum component that is arranged to perform a checksum on the allocable memory block and storing the results of the checksum within the allocable memory block.

14. (Original) The computer-readable medium of Claim 13, further comprising a checksum examination component that is arranged to examine the results of the checksum to determine the presence of memory errors.

15. (Original) The computer-readable medium of Claim 11, wherein the identifier is the return address of the identified routine.

16. (Original) The computer-readable medium of Claim 11, further comprising a pattern component that is arranged to write a memory overwrite detection pattern within the allocable memory block.

17. (Original) The computer-readable medium of Claim 16, wherein the memory overwrite detection pattern is written within an area of the allocable memory block that is used for alignment purposes.

18. (Original) The computer-readable medium of Claim 11, wherein an identifier is generated and stored for a routine that requests the allocable memory block and an identifier is generated and stored for a routine that frees the memory block.

19. (Original) The computer-readable medium of Claim 11, further comprising an indexing component that is arranged to store a heap index for the allocable memory block within the allocable memory block, wherein the heap index points to one of a plurality of heaps.

20. (Canceled).

21. (Currently Amended) A system for tagging an allocable memory block, comprising:

a computer memory that comprises a heap in which allocable memory blocks can be allocated and freed;

a routine identifier that is arranged to determine the identity of a routine performing one of requesting the allocable memory block, requesting the size of the allocable memory block, and freeing the allocable memory block;

an identifier generator that is arranged to generate an identifier for the routine;

a diagnostic tagger that is arranged to store the identifier in the allocable memory block; and

a memory timestamp system that is arranged to store a timestamp within the allocable memory block, wherein the timestamp is configured to indicate ~~indicates~~ a time when ~~one of the requesting and the freeing of the allocable memory block is performed~~:

the requesting of the allocable memory block is performed unless the timestamp indicates a time when the allocable memory block is freed; and

the freeing of the allocable memory block is performed unless the timestamp indicates a time when the allocable memory block is requested,

such that, upon detection of a memory usage error involving the allocable memory block, the identifier for the routine and the timestamp provide information usable in determining whether if the routine is causing memory errors.

22. (Original) The system of Claim 21, further comprising a memory verification system that is arranged to examine the heap to determine the presence of memory errors.

23. (Original) The system of Claim 22, further comprising a memory verification system that is arranged to perform a checksum on the allocable memory block and storing the results of the checksum within the allocable memory block.

24. (Original) The system of Claim 23, further comprising a memory verification system that is arranged to examine the results of the checksum to determine the presence of memory errors.

25. (Original) The system of Claim 21, wherein the identifier is the return address of the identified routine.

26. (Original) The system of Claim 21, further comprising a memory verification system that is arranged to write a memory overwrite detection pattern within the allocable memory block.

27. (Original) The system of Claim 26, wherein the memory overwrite detection pattern is written within an area of the allocable memory block that is used for alignment purposes.

28. (Original) The system of Claim 21, wherein an identifier is generated and stored for a routine that requests the allocable memory block and an identifier is generated and stored for a routine that frees the memory block.

29. (Original) The system of Claim 21, further comprising a memory indexing system that is arranged to store a heap index for the allocable memory block within the allocable memory block, wherein the heap index points to one of a plurality of heaps.

30. (Canceled).